

# The Design of A Reaktor Core Model of the Korg MS-20 VCF

James J. Clark

Sept. 2005

The Korg MS-20 is one of the most popular analog music synthesizers. A good part of its appeal is due to the sound of its filter. Compared to filters such as the Moog ladder filter, the MS-20 filter has a rather harsh sound, especially at high resonant settings. This is due to the (deliberately) extreme non-linearity of its circuitry. In this note I will outline an attempt at modeling the MS-20 filter in Reaktor.

I will take as my starting point the basic form of the MS-20 filter as shown in figure 1, which is taken from the documentation for the Doepfer A-106 filter, which is itself an attempt at reproducing the MS-20 circuit.

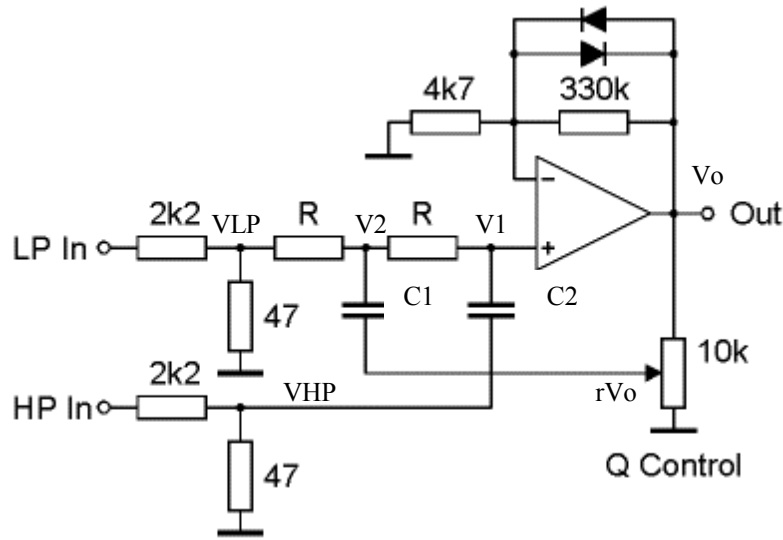


Figure 1. Basic principle of the Doepfer A-106 MS-20 filter (from the Doepfer web site).

Let us begin by analyzing the electronic structure of the circuit. Perhaps the most important aspects of the circuit from the point of view of the filtering action are the capacitors  $C1$  and  $C2$ . In general, capacitors act to integrate the current flowing into them, that is, the voltage across them is given by  $V(t) = V(0) + \frac{1}{C} \int_0^t I(\tau) d\tau$ .

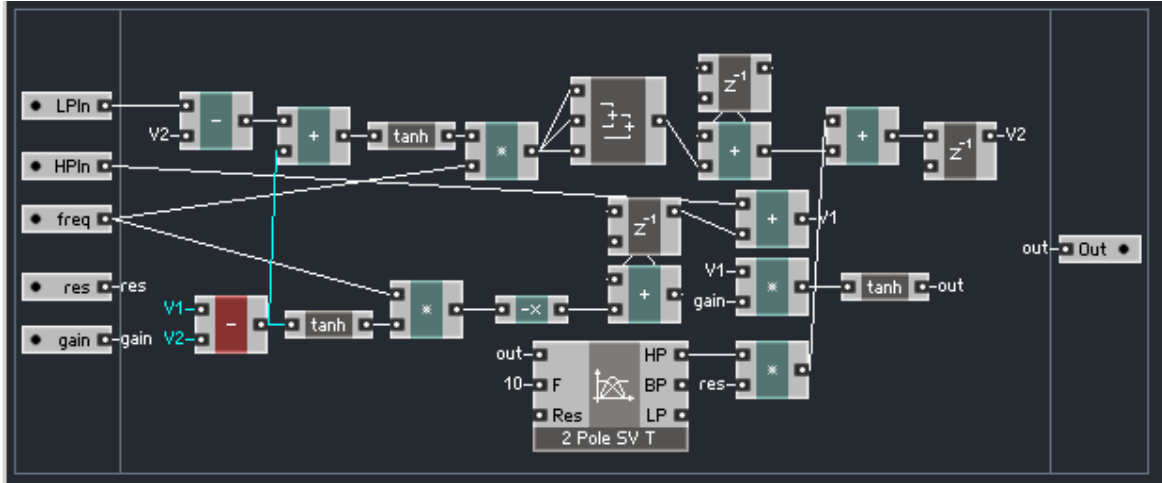
Let us assume that all voltages in the circuit are zero at time  $t=0$ . Then, applying Kirchoff's Current Law for capacitor  $C1$ , we have:

$$V_2 - rV_o = \frac{1}{C_1} \int_0^t I_{C1}(\tau) d\tau = \frac{1}{RC_1} \int_0^t (V_{LP} - 2V_2 + V_1) d\tau$$

and for capacitor  $C2$  we have:

$$V_1 - V_{HP} = \frac{1}{C_2} \int_0^t I_{C2}(\tau) d\tau = \frac{1}{RC_2} \int_0^t (V_2 - V_1) d\tau$$

The amplifier acting on  $V1$  produces the output voltage. Because of the diodes in the amplifier feedback path, the amplifier is nonlinear, so that  $V_o = A(V_1)$ . These equations can be implemented directly in Reaktor, using adders, subtractors, and multipliers, as shown in figure 2.



**Figure 2.** A Reaktor Core implementation of the MS-20 filter using a very simple digital integrator.

A simple first-order digital circuit is used to perform the integration operation. It implements the difference equation

$$y(n) = y(n-1) + Tx(n)$$

where  $T$  is the sampling period. In the diagram shown above the value of  $T$  is absorbed into the value of the *freq* input. This has the side effect that the scaling of the filter cutoff frequency with the *freq* input will depend on the sampling rate. The delay operation (to get  $y(n-1)$  from  $y(n)$ ) is implemented with a  $z^{-1}$  block. This type of integrator can have problems when faced with a signal that has a DC component, as the integrator will quickly saturate. This can be ameliorated by multiplying the feedback (of the delayed output  $y(n-1)$ ) by some value slightly less than 1. In the above circuit it doesn't cause any problem, except on occasion during initialization or startup.

A delay element is added to the output of each integrator. The one for the integrator associated with  $C1$  is the delay element inside the integrator, but the one associated with  $C2$  is inserted after the integrator output is added to the resonance feedback signal. The delay elements are required to prevent loops caused by the feedback of  $V1$  and  $V2$  in the circuit (note that if you tried feeding these signals back without adding in a delay element, Reaktor would do it for you). This makes the  $z$ -transform of the integrator block

effectively equal to  $\frac{z^{-1}}{(1 - z^{-1})}$ .

In the schematic shown above, the scaling of the integrand by the factor  $1/RC$  is implemented by a multiplier (which multiplies the integrand by the input *freq*). The integrand is also passed through a hyperbolic tangent operation (the *tanh* modules). In many analog synthesizers, the effect of the resistors are mimicked using active voltage-controllable devices such as transconductance amplifiers or Vactrol elements. These

allow the resistance value, and hence the filter cutoff frequency, to be electronically controlled. These devices are typically non-linear, however, in that their effect resistance values depend on the voltage across the device. When the voltage is small the variation of resistance is very small, but as the voltage increases the resistance increases greatly. This dependence on the voltage is fairly well modeled by a hyperbolic tangent function. The level of distortion caused by the nonlinear resistors depends on the input levels, and to some extent on the gain level of the output non-linearity. If the input levels are much less than 1, there will be little effect of the resistor non-linearities. So, if you like harsh sounds, crank up the input level!

In the Korg MS-20 filter, the output operational amplifier is deliberately made nonlinear by the addition of diodes in its feedback path. We can also model this non-linearity with a hyperbolic tangent function. The output is scaled by the *gain* input value before passing it through the *tanh* module. This allows external control over the depth of the non-linearity.

A resonant peak is added by feeding back some of the output to the input via the capacitor C1. To avoid problems with saturating the integrator associated with C1 we add a DC blocking high-pass filter to the resonance feedback. Note that this filter will not self-oscillate.

The mx-20 filter that was uploaded (as part of the mx-20 bassline ensemble) to the Reaktor User Library was slightly different than that described above. The modified version is shown in figure 3. It implements the following equation for the capacitor C1:

$$V_2 - rV_o = \frac{1}{C_1} \int_0^t I_{C1}(\tau) d\tau = \frac{1}{RC_1} \int_0^t (V_{LP} - V_2) d\tau$$

This equation implies that there is a voltage buffer inserted between capacitor C1 and the right-hand resistor R. This means that no current flows from C1 into C2. The effect of this change is to reduce the size of the signal level swings, and leads to a somewhat smoother sounding filter (at least to my ears).

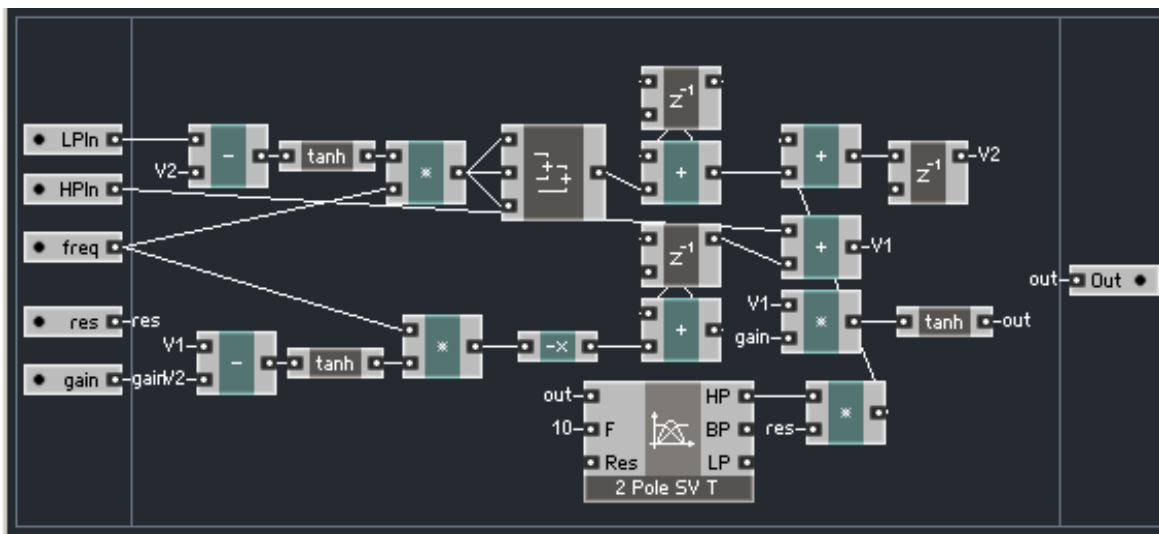


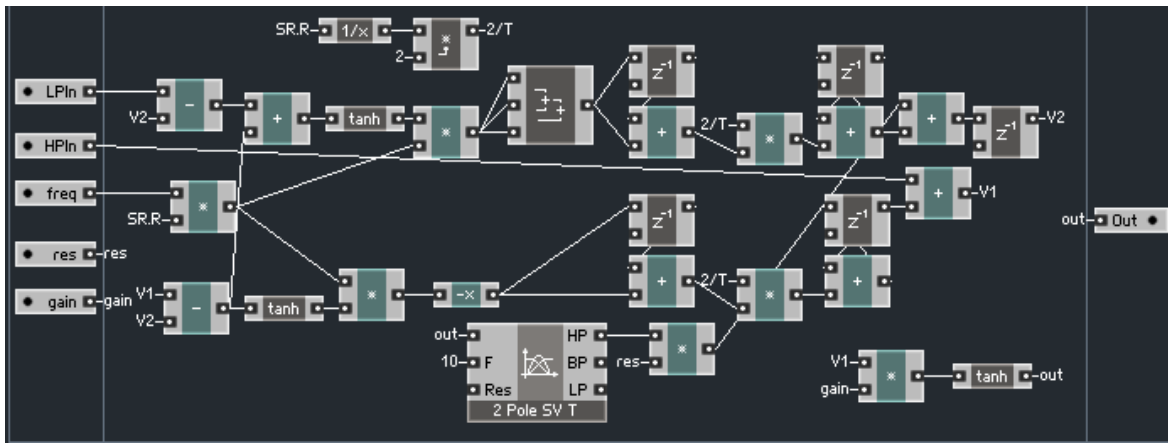
Figure 3. A slight variation on the Reaktor Core implementation of the MS-20 filter.

The integrator used in the above implementations is a very simple approximation to an ideal analog integrator. The most commonly used implementation of a digital integrator is the *bilinear transform* integrator. This uses the trapezoidal approximation to integration, and can be implemented by the difference equation:

$$y(n) = y(n-1) + \frac{T}{2}(x(n) + x(n-1))$$

As in the simple integrator, we need to add an extra delay to the output of the integrator so that we can feed it back. Thus, the z-transform of the integrator is  $\frac{(1+z^{-1})z^{-1}}{(1-z^{-1})}$ .

Using this form for the integrators we get the following implementation of our filter:



**Figure 4. Reaktor Core implementation of the MS-20 filter using a bilinear digital integrator.**

The three different versions of the filter presented in this note all sound somewhat similar, but I prefer the sound of the version shown in figure 3.

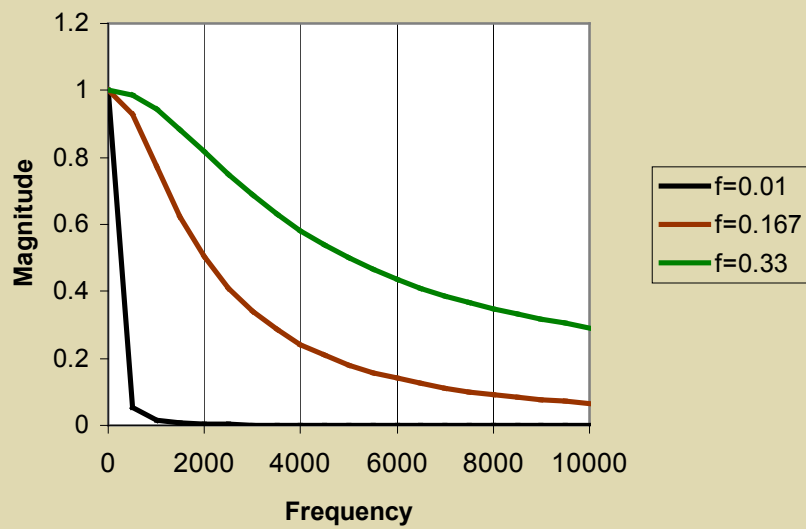
The Lowpass and Highpass transfer functions of the circuit in figure 3 can be shown to be:

$$H_{LP}(z) = \frac{3f^2}{[z - (1-3f)][z - (1-f)]}; \quad H_{HP}(z) = \frac{[z-1]}{z[z - (1-f)]}$$

For the filter to be stable, the poles must be inside the unit circle, therefore we must have that  $0 \leq f \leq 2/3$ . For  $f > 1/3$ , one of the poles of the lowpass filter has a negative value, and the frequency response of the filter begins to rise again at high frequencies, so it is best to keep  $f \leq 1/3$ . The frequency responses can be obtained by evaluating the magnitudes of  $H_{LP}(z)$  and  $H_{HP}(z)$  for  $z = \exp(j\omega T)$ . These are shown in figures 5 and 6 for  $f = 0, 1/6, 1/3$ .

I will leave as an exercise (a difficult one!) the derivation of the frequency responses when the resonance feedback and the circuit nonlinearities are taken into account.

### Lowpass Filter Responses



### Highpass Filter Responses

